RSA Based Consensus Algorithm for Lightweight Private Blockchain Network

Nishant Gupta[1,*], and *Ankit Kumar Jain*[2]
[1,2]Department of Computer Engineering, National Institute of Technology, Kurukshetra

Abstract. Consensus algorithms are essential for achieving agreement among nodes in blockchain systems. However, traditional consensus algorithms such as Proof of Work (PoW) and Proof of Stake (PoS) can be resource-intensive and unsuitable for lightweight private blockchain applications. This paper proposes using the RSA (Rivest–Shamir–Adleman) encryption algorithm as a consensus algorithm for a lightweight private blockchain in the context of a college placement system. RSA offers several advantages over traditional consensus algorithms, including simplicity, efficiency, and security. Moreover, RSA can be implemented on resource-constrained nodes, making it a promising solution for lightweight blockchain applications. The resource constrained nodes are students, academic department, training and placement cell department, and placement cell coordinator. The company acts as a client. The movement of the student's data to companies is recorded as transactions on the distributed ledger or blockchain, allowing the student to track its progress.

1 Introduction

A continuously growing list of records called blocks is maintained by blockchain, which is a distributed ledger. These blocks are linked and protected using cryptography, and each one includes transaction data, a timestamp, and a cryptographic hash of the preceding block. By design, a blockchain resists data modification, providing a secure and transparent way of storing and transferring information. The key features of blockchain are:

- Decentralization: Blockchain is a decentralized technology that operates without a central authority. Instead, it uses a network of nodes to maintain the ledger, ensuring that no single entity controls the data.
- Immutability: Once data is appended to the blockchain, it is immutable and cannot be altered or erased. This feature ensures the integrity and security of the data.
- Transparency: Every transaction that takes place on the blockchain is open and visible to all members of the network, ensuring transparency. Blockchain is an ideal solution for applications where transparency and accountability are crucial.

* Corresponding author: nishant.gupta186@gmail.com

- Security: The cryptographic techniques used in blockchain make it highly secure. Data on the blockchain is protected by complex mathematical algorithms, making it virtually impossible to hack.
- Efficiency: Blockchain enables secure and fast transactions without intermediaries like banks or other financial institutions. This can help reduce costs and increase efficiency in various industries.

Consensus algorithms are an integral part of blockchain technology as they ensure that all nodes in a decentralized network agree on the same version of the distributed ledger. However, the consensus algorithm choice depends on the blockchain's specific use case. For example, a public blockchain like Bitcoin requires a consensus algorithm to withstand attacks from malicious nodes and ensure that only valid transactions are included. On the other hand, a private blockchain used within an organization may prioritize efficiency and privacy over security.

Blockchain technology employs several consensus algorithms, including Proof of Work (PoW) [1], Proof of Stake (PoS) [2], Delegated Proof of Stake (DPoS) [3], and Practical Byzantine Fault Tolerance (PBFT) [4]. Each algorithm has advantages and disadvantages, making them suitable for specific use cases. In Bitcoin, Proof of Work serves as the consensus algorithm, in which miners compete to solve intricate mathematical problems to verify transactions and append them to the blockchain. Validators must hold a specific amount of cryptocurrency as collateral in Proof of Stake, reducing the energy consumption required in PoW. DPoS allows token holders to vote for delegates who validate transactions, while PBFT allows a network to achieve consensus despite the defective nodes. In summary, the choice of consensus algorithm in a blockchain depends on the network's specific requirements, including security, scalability, efficiency, and decentralization.

1.1 Types of Blockchain

Public Blockchains and Private Blockchains are the two main types of blockchain networks.

A public blockchain is an open, decentralized network where anyone can join the network, participate in the consensus process, and read/write data to the blockchain. It is a permissionless network where there are no restrictions on who can participate in the network. Anyone can create a public blockchain and maintain it without any central authority. Bitcoin, Ethereum, and Litecoin are some examples of public blockchains.

A private blockchain is a permissioned network where only a select group of individuals or entities can participate. The network is not open to the public and is not decentralized like public blockchains. Private blockchains are often used by companies and organizations that want to maintain control over their data and keep it private. A central authority usually manages the network; the participants must be authenticated before joining the network. Private blockchains are also called permissioned blockchains. Examples of private blockchains are Hyperledger Fabric and R3 Corda.

Table 1 that summarizes the differences between public and private blockchains. Public blockchains are open to anyone, decentralized, transparent, and immutable. On the other hand, private blockchains are restricted, centralized, partially transparent, and immutable.

Table 1. Comparison of public and private blockchain networks.

Feature	Public Blockchain	Private Blockchain
Access	Open to anyone	Restricted

Decentralization	Fully Decentralized	Partially or fully centralized
Consensus Mechanism	PoW, PoS, etc.	Various, depending on the network
Transparency	Fully transparent	Partially or fully opaque
Data privacy	Publicly visible	Private or partially private
Immutability	Immutable	Immutable
Token Economics	Cryptocurrency or token-based	Can be token based or not
Governance	Decentralized, with no central authority	Centralized, with central authority

The remainder of this paper is organized as follows. Section 2 reviews related work on consensus algorithms for lightweight blockchain. Section 3 presents the proposed RSA consensus algorithm in detail. Section 4 presents the analysis of the RSA as a consensus algorithm. Finally, Section 5 concludes the paper.

2 Related work

Raghav et al. [5] proposed a lightweight consensus algorithm, PoEWAL (Proof of Elapsed Work and Luck), for IoT blockchain applications specifically designed for IoT devices with limited resources. In blockchain applications for IoT, IoT devices participate in the consensus mechanism, and PoEWAL is an energy-efficient and low-latency consensus algorithm that requires less computational power. In PoEWAL, each miner is allotted a specific duration to solve a cryptographic puzzle within a predetermined time window, similar to the proof of work. After the time window, each miner publishes their answer to the network's other miners. The miners then compile and contrast each other's solutions, and the miner with the most consecutive zeros in their solution is declared the winner and updates the block. The miner with the lower nonce value is selected when multiple miners have equal consecutive zeros.

To provide proof of elapsed work, the miner must solve the puzzle within the allocated time. The puzzle is created by concatenating a random number, known as a nonce value, with the hash of the most recent block. The solution to the puzzle is the hash value with the most consecutive zeros starting from the most significant bit. The miner with the most consecutive zeros wins the right to update the blockchain ledger. In the case of a tie, which occurs when two or more miners have an equal number of consecutive zeros, the algorithm chooses the miner with the lower nonce value as the winner. The probability of multiple miners having identical nonce values is low. However, in the unlikely event that this occurs, the algorithm compares the hash values and selects the one with the smallest hash value.

A weighted Byzantine fault tolerance consensus protocol was proposed by Hongwu et al. [6] for use in consortium blockchains. Compared to PBFT, this consensus algorithm provides improved throughput and consensus delay. The WBFT protocol employs a dynamic weighting mechanism to curb malicious behaviors of nodes among the consensus nodes. Following each consensus round, the nodes undergo weight updates, and those whose weight surpasses a particular threshold are permitted to participate in the consensus procedure. This strengthens the security of the blockchain network. Additionally, WBFT optimizes

communication by eliminating the commit step of the PBFT algorithm [7], with no impact on the consensus result.

In their work, Annapurna et al. [8] presented the Zero Trust Model, which mandates that each node must approve transactions before they can be committed. This model ensures the security of shared data by allowing owners to monitor data while various custodians handle it. The consensus algorithm proposed by the authors allows users to place trust in the network, as malicious nodes cannot obtain approval from all nodes, thereby preventing transaction approval. The authors used the college placement system as a use case to demonstrate the proposed consensus algorithm. The algorithm has been extended to implement a decentralized, diversified, and automated placement system. Data transfers between students and companies are recorded as transactions in the distributed ledger or blockchain, enabling students to keep track of their data.

3 Proposed work

The unique contributions of this paper that progress the field of blockchain technology are:

- RSA is proposed as a new consensus algorithm for lightweight blockchain.
- The novel consensus algorithm proves beneficial for distributed systems with limited resources.

To build a model that meets the objectives outlined above, we chose the placement system used in colleges as a use case. Under the current system employed by the placement department, companies visit the campus to recruit students, specify their eligibility criteria, and request information about the students. The limitations arising from this model include the inability to authenticate the information provided by students during the registration stage and the need for access for students to track data.

3.1 RSA cryptographic algorithm

The RSA (Rivest-Shamir-Adleman) [9] is a popular public-key cryptographic algorithm [10] utilized for secure data transmission. It was developed in 1977 and is named after its creators, Ron Rivest, Adi Shamir, and Leonard Adleman.

RSA utilizes the factoring problem of large prime numbers, which states that it is computationally infeasible to factor the product of two large prime numbers into their respective primes. The algorithm uses a pair of keys: a public key, which can be accessible to anyone, and a private key, which is kept confidential by the owner.

In the RSA encryption scheme, the sender encrypts a message with the recipient's public key, and only the recipient's private key can decrypt the message. This guarantees that only the intended recipient can read the message since they possess the private key required to decrypt the message.

The RSA can also be utilized to create digital signatures. In this process, the sender signs a message with their private key, and the recipient confirms the signature's validity using the sender's public key. This technique ensures non-repudiation and authentication, as the recipient can verify that the claimed sender indeed sent the message.

RSA has been widely used in various applications such as secure email, SSL/TLS [11], and digital signatures [12]. It is considered one of the most secure and reliable cryptographic algorithms [13]. However, it is computationally expensive and can be slow for large messages, so it is typically used for encrypting only small amounts of data or generating digital signatures.

3.2 RSA as a proposed consensus algorithm

The paper proposes RSA as a novel consensus algorithm to make blockchain lightweight and compatible with nodes with limited resources. The process commences with individual participants/nodes generating transactions (Trx) that include data and grouping them into a block. Subsequently, nodes disseminate the blocks for assessment and validation by trusted nodes within the network. To ensure secure block transmission, the network user first provides their public key (PuK) to the network and uses their private key (PrK) to sign the block. Validation of the block requires the involvement of trusted nodes within the network. Once a trusted node receives the block, it locates the source's public key and verifies the signature using asymmetric cryptography. This process helps to prevent attackers from extracting the private key. After validation, trusted nodes disseminate the verified blocks with trusted identification, and other nodes must confirm the trusted information before adding blocks to the chain. Upon approving a valid block, the node calculates a hash value (not inverse hash) to connect the next block, as illustrated in Figure 1. All nodes in the network adhere to this principle to maintain the chain. Algorithm 1 outlines the technical steps in implementing RSA as a consensus algorithm.

Algorithm 1: The Proposed RSA as a consensus algorithm.

Inputs : Individual nodes in the network utilize public (PuK) and private keys (PrK), while all nodes adopt SHA-256 hash.

Outputs: Blocks that have been validated and approved are appended to the blockchain.
Step-1 Nodes form blocks with the data.
Step-2 Nodes use their private key to sign and disseminate blocks to the network.
Step-3 The signature is verified by the trusted node using the corresponding public key.
Step-4 If the verification is successful **then**
- The trusted node broadcasts the verified block to the network.
- If the nodes receive the block from a trusted node, they append it to the blockchain.

Step-5 Else
- If the block is not verified, it will be discarded.

Step-6 Return to step 1 to move on to the next block.

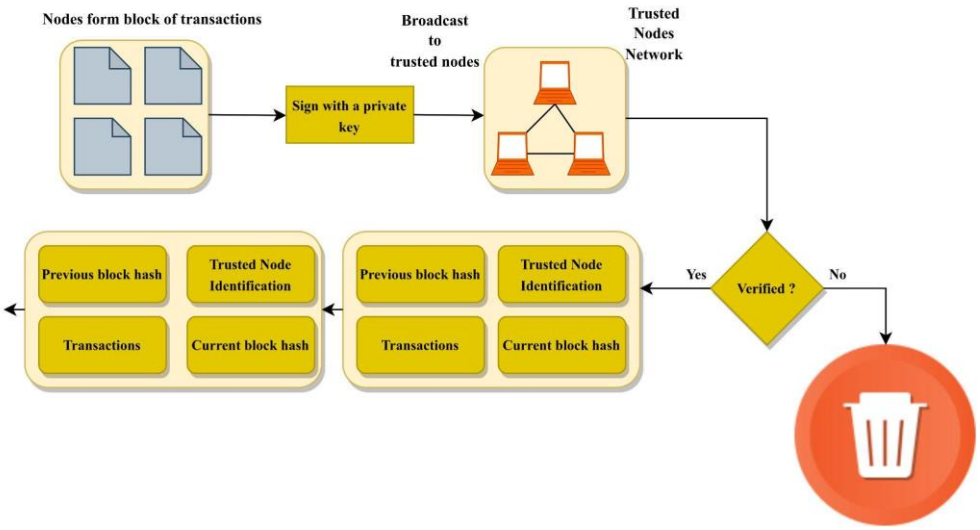

Fig. 1. The proposed RSA as a consensus algorithm.

3.3 The proposed placement system model

The proposed placement system model's process is illustrated in Figure 2. There are five nodes in the proposed placement system model. They are the placement cell coordinator (PCC), company, student, academic department (AD), and training and placement cell department (TNP). The AD and TNP act as trusted nodes. The company acts as a client. The company contacts the placement cell coordinator (PCC), specifying the job role and eligibility criteria and requesting data from interested students. The placement cell coordinator sends an animal to students with a registration link. Interested students click on the registration link and form a block with data. The student nodes sign the blocks with their private key before transmitting the signed block to the academic department. The academic department verifies the student's data with the student's public key and college database. After this, the academic department forms a block with the student's data and his unique ID and signs with its private key and transmits the signed block to the training and TNP for further evaluation. The TNP verifies the signed block with the public key of the AD and college database. After successful verification, the training and the TNP forms a block with the student's data and his unique ID and signs the block with its private key before broadcasting the block to the network. The TNP department also sends back student data to the company. Suppose individual nodes hear the block coming from the TNP department (individual nodes verify the block with the public key of the TNP department). In that case, individual nodes compute the block's hash (not inverse hash) to add it to their local ledger. In this model, the students can track their data and be sure whether their data has been sent to the company.

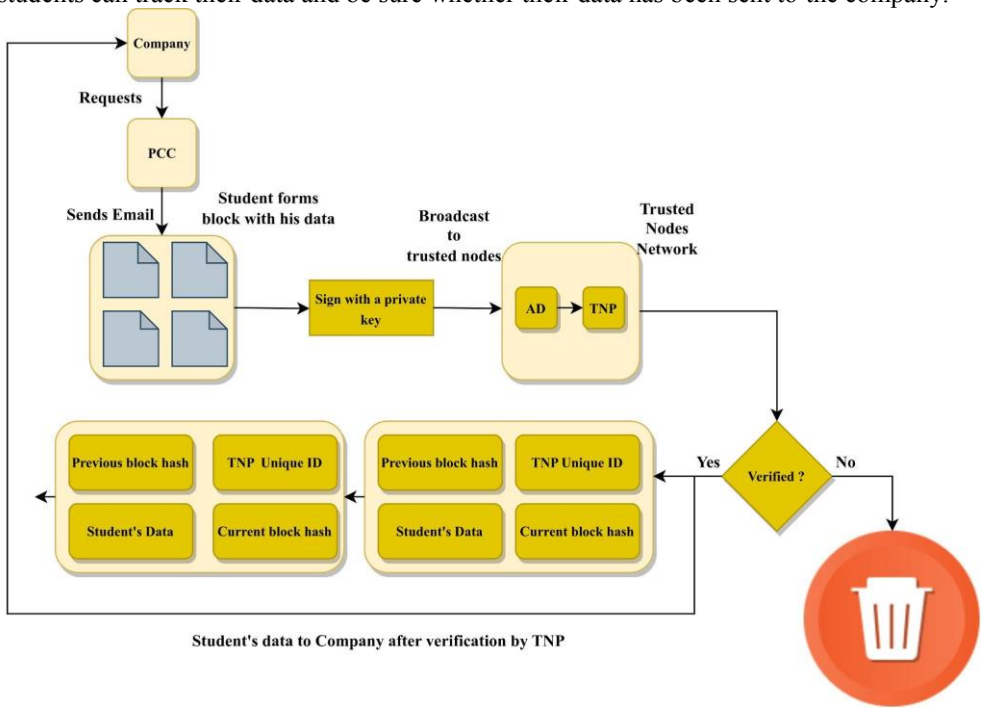

Fig. 2. The workflow of the proposed placement model system.

4 Analysis of the RSA as a consensus algorithm

Using RSA as a consensus mechanism can be advantageous for nodes with limited resources, as it requires less computational power and energy compared to other mechanisms. To

implement RSA as a consensus mechanism, the block contains the necessary information. When someone starts a transaction, a trusted node receives it and starts the validation process. Once the validation has been completed, the trusted node includes the block to their blockchain and distributes it to other nodes. If the trusted node sends the block, other nodes will include it to their local blockchain ledger database. These actions lead to the following assertions:

Assertion - 1: The RSA consensus mechanism requires only a small amount of resources to validate blocks.

Proof: The PoW consensus algorithm is a conventional method deployed in blockchain networks. Individual nodes create data blocks and are verified by miners before being included in the chain. Mining requires computing the hash's inverse, which consumes as much energy as two households do in a day. This high energy consumption makes it infeasible for nodes with limited resources. RSA has been proposed as an alternative consensus mechanism that can evaluate blocks using minimal energy to address this issue. RSA utilizes a digital signature process instead of the computation of an inverse hash. When it comes to cryptography, hash computation (not the inverse hash) and digital signature are both efficient and consume minimal energy [14,15].

Assertion - 2: Compared to PoW, RSA as a consensus mechanism takes significantly less time.

Proof: From the above assertion, PoW consumes a substantial amount of energy during block validation and takes around 10 minutes to permanently confirm a block in the blockchain. This is not viable for lightweight blockchain applications, and a block evaluation time of 10 minutes is deemed unacceptable. RSA is presented as a solution to this problem by enabling block evaluation in a minimal amount of time. Since RSA validates blocks through the use of authentication, we know that considerably less time is required for the cryptographic authentication process.

5 Conclusions

RSA as a consensus algorithm for lightweight blockchain is proposed in this paper, which is a vital component in eliminating centralized dependencies for resource-constrained distributed systems. The proposed consensus algorithm is thoroughly analyzed to validate its effectiveness, demonstrating its effectiveness in addressing the challenges of lightweight blockchain. Through analysis, we have demonstrated that our proposed RSA-based consensus algorithm outperforms traditional consensus algorithms, such as PoW, in terms of efficiency and resource utilization. Additionally, the proposed RSA-based consensus algorithm offers a practical solution for the college placement system, facilitating secure and efficient placement operations.

In summary, using RSA as a consensus algorithm offers significant benefits for private blockchain networks, including improved efficiency, security, and scalability. We hope this research will pave the way for adopting RSA as a consensus algorithm for various private blockchain applications and contribute to developing more efficient and secure consensus algorithms.

References

1. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review*, 21260.
2. King, S., & Nadal, S. (2012). Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August*, *19*(1).

3. Schuh, F., & Larimer, D. (2015). Bitshares 2.0: Financial smart contract platform. *Bitshares Financ. Platf*, *12*.
4. Castro, M., & Liskov, B. (1999, February). Practical byzantine fault tolerance. In *OsDI* (Vol. 99, No. 1999, pp. 173-186).
5. Andola, N., Venkatesan, S., & Verma, S. (2020). PoEWAL: A lightweight consensus mechanism for blockchain in IoT. *Pervasive and Mobile Computing*, *69*, 101291.
6. Qin, H., Cheng, Y., Ma, X., Li, F., & Abawajy, J. (2022). Weighted byzantine fault tolerance consensus algorithm for enhancing consortium blockchain efficiency and security. *Journal of King Saud University-Computer and Information Sciences*, *34*(10), 8370-8379.
7. Hao, X., Yu, L., Zhiqiang, L., Zhen, L., & Dawu, G. (2018, May). Dynamic practical byzantine fault tolerance. In *2018 IEEE conference on communications and network security (CNS)* (pp. 1-8). IEEE.
8. Patil, A. P., Karkal, G., Wadhwa, J., Sawood, M., & Reddy, K. D. (2020, December). Design and implementation of a consensus algorithm to build zero trust model. In *2020 IEEE 17th India Council International Conference (INDICON)* (pp. 1-5). IEEE.
9. Milanov, E. (2009). The RSA algorithm. *RSA laboratories*, 1-11.
10. Patil, P., Narayankar, P., Narayan, D. G., & Meena, S. M. (2016). A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish. *Procedia Computer Science*, *78*, 617-624.
11. Elgohary, A., Sobh, T. S., & Zaki, M. (2006). Design of an enhancement for SSL/TLS protocols. *computers & security*, *25*(4), 297-306.
12. Serhrouchni, A., & Hajjeh, I. (2006, June). Integration of the digital signature in the protocol SSL/TLS. In *Annales Des Télécommunications* (Vol. 61, pp. 522-541). Springer-Verlag.
13. Abood, O. G., & Guirguis, S. K. (2018). A survey on cryptography algorithms. *International Journal of Scientific and Research Publications*, *8*(7), 495-516.
14. Potlapally, N. R., Ravi, S., Raghunathan, A., & Jha, N. K. (2005). A study of the energy consumption characteristics of cryptographic algorithms and security protocols. *IEEE Transactions on mobile computing*, *5*(2), 128-143.
15. Bada, A. O., Damianou, A., Angelopoulos, C. M., & Katos, V. (2021, July). Towards a green blockchain: A review of consensus mechanisms and their energy consumption. In *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)* (pp. 503-511). IEEE.